



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/731,620	12/08/2003	Janaki Krishnaswamy	021756-003100US	4734
51206	7590	12/08/2008	EXAMINER	
TOWNSEND AND TOWNSEND AND CREW LLP TWO EMBARCADERO CENTER 8TH FLOOR SAN FRANCISCO, CA 94111-3834		PHAM, HUNG Q		
		ART UNIT		PAPER NUMBER
		2169		
		MAIL DATE		DELIVERY MODE
		12/08/2008		PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/731,620	KRISHNASWAMY ET AL.	
	Examiner	Art Unit	
	HUNG Q. PHAM	2169	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 28 August 2008.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1 and 8-16 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1 and 8-16 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) Notice of Informal Patent Application

6) Other: _____.

DETAILED ACTION

Response to Amendment

Claim Rejections - 35 USC § 101

The amendment of claim 14 has not complied with 35 U.S.C. § 101. The system as recited in claim 14 is directs to a system comprising software per se. Software per se is not one of the four categories of invention. Software per se is not a series of steps or acts and thus is not a process. Software per se is not a physical article or object and as such is not a machine or manufacture. Software per se is not a combination of substances and therefore is not a composition of matter.

Claim Rejections - 35 USC § 112

The rejection under 35 U.S.C. § 112, second paragraph, has been withdrawn in view of the amendment.

Response to Arguments

1. Applicant's arguments regarding claims 8-14 under 35 U.S.C. § 102 filed 08/28/08 have been fully considered but they are not persuasive.

- As argued by applicant (Remarks, Pages 7-8):

Applicants submit that several of the features recited in claim 1 are not taught by Wall. For example, claim 8 specifically recites a "method for object model design and validation," and "creating an instance of a meta metadata object of an object model in response to user input." Applicants submit that this feature recited in claim 8 is not taught by Wall.

... When a user makes a selection through the GUI, there is no object that is being generated. As such, Wall fails to teach "creating an instance of a meta metadata object of an object model in response to user input."

The examiner respectfully disagrees.

In response to applicant's arguments, the recitation *method for object model design and validation* has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

In order to collect data from user, a GUI with field names and input fields is represented (Wall, Col. 2 Lines 1-15). As shown in FIG. 5, an instance of a person object with inputted personal data is created by inputting information into data fields. The person object instance comprises inputted personal data, e.g., name, address, city and state. As further disclosed by Wall, a class derived from the model class such as a person model may be used to represent a person who has personal data, such as name, address... Nested within the person model class are four objects of the StringFieldModel class (name, address, city and state) that correspond to input fields of a GUI created by the View. Objects and classes of the Model may be designed to represent abstractions of business entities, e.g., customers (Wall, Col. 7 Lines 14-35). The model class as taught by Wall is considered as being equivalent to the claimed *object model*. The person model class comprising four objects of the StringFieldModel from model class is considered as being equivalent to the claimed *meta metadata object*. The person object instance with inputted personal data, e.g., name, address, city and state, is considered as being equivalent to the claimed *instance*. In short, the Wall teaching reads on the claimed *creating an instance of a meta metadata object of an object model*, e.g., an instance of the person model class

comprising four objects of the StringFieldModel of the model class, *in response to user input*, e.g., the person object instance is created by inputting information into data fields.

- As argued by applicant (Remarks, Pages 8-9):

Applicants further submit that Wall fails to teach "if a user selects validation of the object instance, applying one or more completeness validation rules to the object instance." Wall teaches that the View enforces the constraints on input fields in the GUI. (Wall, col. 9, lines 6-9). The user does not have any control over enforcing constraints. Instead, the user's actions themselves are constrained by the rules. The user in Wall does not select to have objects validated or to have constraints applied to the objects. As such, Wall fails to teach "if a user selects validation of the object instance, applying one or more completeness validation rules to the object instance," as recited in claim 8.

The examiner respectfully disagrees.

As shown in FIG. 5, the “Enter” button is used to signal that data entry is completed (Wall, Col. 8 Lines 23-25). As disclosed by Wall, input field constraints relating input field length, input field range of acceptable values including constraints that requires the user to enter data in each input field or not enter data into any of the input fields or at least one nominated input field. When the user fails to adhere to a data validation rule associated with a constrained input field, a pop up window will appear (Wall, Col. 9 Lines 4-27). Thus, if the user clicks the “Enter” button to signal that data entry is completed, the person object instance is validated against input field constraints such as constraint relating input field length, or input field range of acceptable values, or constraints that requires the user to enter data in each input field.

The Wall teaching indicates the claimed limitation *if a user selects validation of the object instance*, e.g., “Enter” button to signal that data entry is completed and trigger the input constraints against the input field, *applying one or more completeness validation rules to the object instance*, e.g., constraints relating input field length, input field range of acceptable values

including constraints that requires the user to enter data in each input field are applied to the object instance.

- As argued by applicant (Remarks, Page 9):

Applicants submit that Wall fails to teach "automatically applying both the one or more correctness validation rules and the one or more completeness validation rules to the object instance prior to deployment of the object instance at runtime." As previously mentioned, Wall does not describe that object models can be designed or validated. The processes as described by Wall occur during runtime of the IMVC software application. (Wall, col. 10, lines 5-8). Accordingly, Wall does not teach "automatically applying both the one or more correctness validation rules and the one or more completeness validation rules to the object instance prior to deployment of the object instance at runtime."

The examiner respectfully disagrees.

In order to enforce the data validation rule associated with the “State” input field as in FIG. 5, a drop-down list box is created. The drop-down list box allows the user to only select from a set of possible choices (CA, CO, JY, NY or PA). The Wall teaching indicates the step of *automatically applying one or more correctness type validation rules to the object instance*, e.g., the constraint of State field is automatically applied to the person object instance, *by confirming the object instance complies with the one or more correctness type validation rules*, e.g., the constraint of State field is applied by confirming the person object instance complies with only the choice from a set of possible choices.

As discussed above, the Wall teaching indicates the claimed limitation *if a user selects validation of the object instance*, e.g., “Enter” button to signal that data entry is completed and trigger the input constraints against the input field, *applying one or more completeness validation rules to the object instance*, e.g., constraints relating input field length, input field range of acceptable values

including constraints that requires the user to enter data in each input field are applied to the object instance.

As further disclosed by Wall, the validation rules are applied to the person object instance to ensure that entered data complies with the validation rules. If the entered data of person object instance is not complied with the validation rules, pop-up windows are displayed for further correction (Wall, Col. 2 Lines 16-46).

In short, the applying of the constraint of State field and constraints relating input field length, input field range of acceptable values as discussed above indicates the step of *automatically applying both the one or more correctness validation rules and the one or more completeness validation rules to the object instance*. The application of the constraint rules is *prior to deployment of the object instance at runtime*, e.g., prior to collecting the personal object instance with corresponding personal data from user at runtime.

2. Applicant's arguments regarding the rejection under 35 U.S.C. § 102 filed 08/28/08 have been fully considered but they are not persuasive.

- As argued by applicant (Remarks, Page 10):

Wall makes no mention or suggestion of object model design and validation. Furthermore, Wall does not describe "creating an instance of a meta metadata object of an object model in response to user input." Wall describes that the View can create a derivative Person Model Class that corresponds to input fields of the GUI. (Wall, col. 7, lines 13-18). The input fields that are presented in the GUI enforce the data validation rules. (Wall, col. 6, lines 50-57). When a user makes a selection through the GUI, there is no object that is being generated. As such, Wall fails to teach "creating an instance of a meta metadata object of an object model in response to user input." Along similar rationale, Applicants submit that Wall does not teach or suggest "a database for storing objects corresponding to an object model and for storing metadata objects describing the object model while designing the object model," as recited in claim 1. The Examiner asserts that Wall shows a database that stores Person model objects, which teaches this feature. (Office Action, p. 4). Applicants respectfully disagree. The database in Wall is not a database that is used during a design phase of an object model. Applicants direct the Examiner to paragraphs [007] and [009]

of the application as filed for exemplary discussion of designing an object model. Likewise, Wall also fails to describe a validation engine as recited in claim 1. As previously discussed, Wall makes no mention or suggestion of validating metadata objects. Instead, Wall describes validation of data.

The examiner respectfully disagrees.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *creating an instance of a meta metadata object of an object model in response to user input...* *Applicants direct the Examiner to paragraphs [007] and [009] of the application as filed for exemplary discussion of designing an object model*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In order to collect data from user, a GUI with field names and input fields is represented (Wall, Col. 2 Lines 1-15). As shown in FIG. 5, an instance of a person object with inputted personal data is created by inputting information into data fields. The person object instance comprises inputted personal data, e.g., name, address, city and state. As further disclosed by Wall, a class derived from the model class such as a person model may be used to represent a person who has personal data, such as name, address... Nested within the person model class are four objects of the StringFieldModel class (name, address, city and state) that correspond to input fields of a GUI created by the View. Objects and classes of the Model may be designed to represent abstractions of business entities, e.g., customers (Wall, Col. 7 Lines 14-35).

As shown in FIG. 5, the *database* at the location specified by the URL is *for storing objects*, e.g., instances of person object with corresponding personal information, *corresponding to an object model*, e.g., person model, *and metadata objects describing the object model*, e.g., name object, address object, city object and state object, *while designing the object model*, e.g., name object, address

object, city object and state object are stored while designing the personal model using the class.

As specified in the previous Office Action, the illustration at Col. 5 Lines 10-27 from Wall reference indicates *a validation engine*, e.g., the controller.

Applicant's argument, e.g., *Wall makes no mention or suggestion of validating metadata objects*, have been fully considered but they are not persuasive. The claimed limitation has been amended. The amended limitation *a validation engine for validating the metadata objects stored in the database by confirming the metadata objects comply with one or more validation rules* was not described in the Specification. In light of the Specification of the current application, a set of deployed metadata is validated (Paragraph 0048). Nowhere in the Specification describes the stored metadata is validated. Therefore, the examiner respectfully declines to answer to this argument. A new ground of rejection of this limitation is made as in the following manners.

- As argued by applicant (Remarks, Page 10):

Applicants submit that the combination of Wall and Raghuvir fails to teach or suggest "a configuration management module for creating a deployable collection of metadata objects from the metadata objects stored in the database, wherein the deployable collection represents a tree of metadata objects starting at a root metadata object." The Examiner recognizes that Wall does not teach this feature. (Office Action, p. 5). Raghuvir is relied upon for such teaching.

The examiner respectfully disagrees.

As disclosed by Wall, *a deployable collection of objects*, e.g., name object, address object, city object and state object are created from the group of name object, address object, city object and state object in the database as discussed above.

The missing of Wall is *a tree of metadata objects starting at a root metadata object* that represents *the deployable collection*.

Raghuvir teaches *a tree of metadata objects starting at a root metadata object* (FIG. 6).

A tree structure as taught by Raghuvir could be used to represent the name object, address object, city object and state object. By using the tree, the relationship between objects is enforced when objects are incorporated into a class.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 14 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 14 directs to a system comprising software per se. Software per se is not one of the four categories of invention. Software per se is not a series of steps or acts and thus is not a process. Software per se is not a physical article or object and as such is not a machine or manufacture. Software per se is not a combination of substances and therefore is not a composition of matter.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claim 1 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Regarding claim 1, the claimed limitation *a validation engine for validating the metadata objects stored in the database by confirming the metadata objects comply with one or more validation rules* was not described in the Specification.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 1 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding claim 1, the clauses *the validation subject* (Line 15), *the subject* (Line 16) reference to other items in the claim. It is unclear what item is being referenced.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1, 15 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wall et al. [USP 7,028,288 B2] in view of Raghuvir et al. [US 2004/0249823 A1].

Regarding claim 1, Wall teaches a system for object model design and validation, the system comprising:

a client device configured to receive user input and provide a user interface to a user (Wall, Col. 4 Lines 23-36);

a database for storing objects corresponding to an object model and metadata objects describing the object model while designing the object model (In order to collect data from user, a GUI with field names and input fields is represented (Wall, Col. 2 Lines 1-15). As shown in FIG. 5, an instance of a person object with inputted personal data is created by inputting information into data fields.

The person object instance comprises inputted personal data, e.g., name, address, city and state. As further disclosed by Wall, a class derived from the model class such as a person model may be used to represent a person who has personal data, such as name, address... Nested within the person model class are four objects of the StringFieldModel class (name, address, city and state) that correspond to input fields of a GUI created by the View. Objects and classes of the Model may be designed to represent abstractions of business entities, e.g., customers. Objects such as the four objects of the StringFieldModel class may exist separately from the View (Wall, Col. 7 Lines 14-35). As shown in FIG. 5, the *database* at the location specified by the URL is *for storing objects*, e.g., instances of person object with corresponding personal information, *corresponding to an object model*, e.g., person model, *and metadata objects describing the object model*, e.g., name object, address object, city object and state object, *while designing the object model*, e.g., name object, address object, city object and state object are stored while designing the personal model using the class);

a configuration management module for creating a deployable collection of metadata objects from the metadata objects stored in the database (*a deployable collection of objects*, e.g., name object, address object, city object and state object are created from the group of name object, address object, city object and state object in the database as discussed above)

a validation engine for validating the metadata objects stored in the database by confirming the metadata objects comply with one or more validation rules (This limitation was not described in the Specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of this feature)

wherein said validation engine is configured to perform completeness validation on the deployable collection in response to a user entered command to perform validation on the validation subject (Wall, Col. 9 Lines 5-27),

to automatically perform correctness validation on the deployable collection when the subject is created or updated (Wall, Col. 8 Lines 34-60), and

to automatically perform completeness and correctness validation on the deployable collection when requested by the configuration management module (Wall, Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

The missing of Wall is the claimed limitation *a tree of metadata objects starting at a root metadata object* to represent the name object, address object, city object and state object.

Raghuvir teaches *a tree of meta objects starting at a root meta object* (FIG. 6).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to include the hierarchical collection as taught by Raghuvir into Wall technique in order to have a representation of the collection and enforce the relationship between objects when incorporating into a class.

Regarding claim 15, Wall and Raghuvir, in combination, teach all of the claimed subject matter as discussed above with respect to claim 1, Wall further discloses *a deployment manager to deploy the validated metadata objects during runtime* (Wall, Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Regarding claim 16, Wall and Raghuvir, in combination, teach all of the claimed subject matter as discussed above with respect to claim 1, Wall further discloses *after applying both the one or more correctness validation rules and the one or more completeness validation rules, deploying the object instance during runtime* (Wall, Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 8-14 are rejected under 35 U.S.C. 102(e) as being anticipated by Wall et al.

[USP 7,028,288 B2].

Regarding claims 8 and 14, Wall teaches a system and computer-implemented method for object model design and validation, the method and system comprising:

a database for storing objects and for storing meta metadata objects of an object model while designing the object model (In order to collect data from user, a GUI with field names and input fields is represented (Wall, Col. 2 Lines 1-15). As shown in FIG. 5, an instance of a person object with inputted personal data is created by inputting information into data fields. The person object instance comprises inputted personal data, e.g., name, address, city and state. As further disclosed by Wall, a class derived from the model class such as a person model may be used to represent a person who has personal data, such as name, address... Nested within the person model class are four objects of the StringFieldModel class (name, address, city and state) that correspond to input fields of a GUI created by the View. Objects and classes of the Model may be designed to represent abstractions of business entities, e.g., customers. Objects such as the four objects of the StringFieldModel class may exist separately from the View (Wall, Col. 7 Lines

14-35). As shown in FIG. 5, the *database* at the location specified by the URL is *for storing objects*, e.g., instances of person object with corresponding personal information *and meta metadata objects*, e.g., name object, address object, city object and state object, *of an object model*, e.g., person model, *while designing the object model*, e.g., name object, address object, city object and state object are stored while designing the personal model using the class);

creating an instance of a meta metadata object of an object model in response to user input (In order to collect data from user, a GUI with field names and input fields is represented (Wall, Col. 2 Lines 1-15). As shown in FIG. 5, an instance of a person object with inputted personal data is created by inputting information into data fields. The person object instance comprises inputted personal data, e.g., name, address, city and state. As further disclosed by Wall, a class derived from the model class such as a person model may be used to represent a person who has personal data, such as name, address... Nested within the person model class are four objects of the StringFieldModel class (name, address, city and state) that correspond to input fields of a GUI created by the View. Objects and classes of the Model may be designed to represent abstractions of business entities, e.g., customers (Wall, Col. 7 Lines 14-35). The model class as taught by Wall is considered as being equivalent to the claimed *object model*. The person model class comprising four objects of the StringFieldModel from model class is considered as being equivalent to the claimed *meta metadata object*. The person object instance with inputted personal data, e.g., name, address, city and state, is considered as being equivalent to the claimed *instance*. In short, the Wall teaching reads on the claimed *creating an instance of a meta metadata object of an object model*, e.g., an instance of the person model class comprising four objects of the StringFieldModel of the model class, *in response to user input*, e.g., the person object instance is created by inputting information into data fields);

automatically applying one or more correctness type validation rules to the object instance by confirming the object instance complies with the one or more correctness type validation rules (Wall, Col. 8 Lines 34-60);

if a user selects validation of the object instance, applying one or more completeness validation rules to the object instance (As shown in FIG. 5, when the user select Enter, the completeness validation rules are applied (Col. 9 Lines 5-27)); and

automatically applying both the one or more correctness validation rules and the one or more completeness validation rules to the object instance prior to deployment of the object instance at runtime (Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Regarding claim 9, Wall teaches all of the claimed subject matter as discussed above with respect to claim 8, Wall further discloses *the meta metadata object is one of an attribute, an association, an object and a collection of objects* (Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Regarding claim 10, Wall teaches all of the claimed subject matter as discussed above with respect to claim 8, Wall further discloses *the meta metadata object is an association and wherein the object instance to which a validation rule is applied includes the two objects associated by the association* (FIG. 5, Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Regarding claim 11, Wall teaches all of the claimed subject matter as discussed above with respect to claim 8, Wall further discloses the step of *automatically applying the one or more correctness type validation rules to the instance if the instance is automatically updated or manually updated* (FIG. 5, Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Regarding claim 12, Wall teaches all of the claimed subject matter as discussed above with respect to claim 11, Wall further discloses *the meta metadata object is one of an attribute and an object* (Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Regarding claim 13, Wall teaches all of the claimed subject matter as discussed above with respect to claim 8, Wall further discloses *the meta metadata object is one of an aggregated collection and a deployable collection* (Col. 8 Lines 34-60 and Col. 9 Lines 5-27).

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HUNG Q. PHAM whose telephone number is 571-272-4040. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JAMES K. TRUJILLO can be reached on 571-272-3677. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/HUNG Q. PHAM/
Primary Examiner
Art Unit 2169

November 26, 2008